

Bilinear Vector Warps

①

- Use Einstein summation convention:

repeated index means sum over that one, as in

$$v_i w_i = \sum_i v_i w_i = \text{dot product of 2 vectors } \underline{v} \cdot \underline{w}$$

$$A_{ij} x_j = \sum_j A_{ij} x_j = \text{matrix-vector multiply of } \underline{A} \underline{x}$$

- Use $[[-]]_{ij}$ to indicate matrix formed by the contents, at-subscript ij .

- Use A_{ij}^{-1} to mean the ij element of matrix \underline{A} , not

the reciprocal of A_{ij} !

- $\delta_{ij} = \text{Kronecker delta} = \begin{cases} 1 & \text{when } i=j \\ 0 & \text{when } i \neq j \end{cases} = \text{identity matrix}$

- Bilinear warp in 3D is a generalization of the 1D function $y(x) = \frac{ax+b}{1+cx} \Rightarrow x = \frac{\frac{1}{a}y - \frac{b}{a}}{1 - \frac{c}{a}y}$

Showing the inverse of a bilinear warp in 1D is also bilinear.

- In 3D, we write

$$y_i = \left[\left[\delta_{pq} + c_{pqr} x_r \right]_{ij}^{-1} \left[A_{jkr} x_k + b_j \right] \right]$$

where c_{pqr} is a 3-index object, which when multiplied against vector \underline{x} reduces to a 2-index (object) which can be inverted. So there are $3^3 + 3^2 + 3 = 39$ parameters.

$$\begin{matrix} \underbrace{3^3}_{c_{pqr}} & \underbrace{3^2}_{A_{jk}} & \underbrace{3}_{b_j} \\ & & \end{matrix} = 39 \text{ parameters.}$$

Multiplying out by the matrix $[[-]]$ on both sides: ②

$$\textcircled{A} \quad [S_{ji} + C_{jik} x_k] y_i = A_{jk} x_k + b_j$$

notice that indices on the left have been adjus to match those on the right.

or

$$\textcircled{A'} \quad y_j + C_{jik} x_k y_i = A_{jk} x_k + b_j$$

or

$$y_j - b_j = [A_{jk} - C_{jik} y_i] x_k$$

multiply both sides by A^{-1} in the form A_{lj}^{-1}

$$\textcircled{AA'} \quad A_{lj}^{-1} [y_j - b_j] = [S_{lk} - A_{lj}^{-1} C_{jik} y_i] x_k$$

Now note that \textcircled{AA} and \textcircled{A} have the same form, properly considered.

The RHS of \textcircled{A} is a matrix-vector multiply of x plus an offset vector. The LHS of \textcircled{AA} has that form with y multiplied by A^{-1} and the offset vector being $-A^{-1} b$.

The LHS of \textcircled{A} multiplies y by the identity matrix plus a matrix that is a linear function of x . The RHS of \textcircled{AA} is similar. To make the parallels clearer, and to see how to compute (in detail) the elements of the inverse warp, relabel the indices in \textcircled{AA} to match those in \textcircled{A} :

Those in \textcircled{A} :
 $l \rightarrow j$ [the "free" index]
 $j \rightarrow k$ [only on LHS of \textcircled{AA}]
 $l \leftrightarrow j$ [RHS of \textcircled{AA}]
 $k \leftrightarrow i$ [RHS of \textcircled{AA}]

$$\textcircled{AA'} \quad A_{jk}^{-1} y_k - A_{jk}^{-1} b_k = [S_{ji} - A_{jk}^{-1} C_{kri} y_k] x_i$$

Comparing $\textcircled{AA'}$ to \textcircled{A} , and writing \sim over the parameters for the inverse warp, we see

$$\tilde{b}_j = -A_{jk}^{-1} b_k \quad \tilde{A}_{jk} = A_{jk}^{-1}$$

$$\tilde{C}_{jik} = -A_{jl}^{-1} C_{lki} \quad \left[\begin{array}{l} \text{Apply } A^{-1} \text{ matrix to 1st index} \\ \text{of } C \text{ and transpose last} \\ \text{2 indices} \end{array} \right]$$

Shifting

In the code, the actual bilinear transform is done

(3)

like so:

$$y_i = \left[\left[\delta_{pq} + f \cdot C_{pqr} (x_r - \Delta_r) \right] \right]_{ij}^{-1} [A_{jk} x_k + b_j]$$

where $\underline{A} \hat{=} \underline{b}$ are stored in a 4x4 (mat44) struct, and the vector $\underline{\Delta}$ is the center of coordinates of all voxels being transformed and f is a fixed scale factor that (in 3dAllineate) is calculated as $f = 1.2 / \max[L_x, L_y, L_z]$ where $L_p =$ half-width of bounding box of data in the p^{th} coordinate dimension.

The reason for centering at $\underline{\Delta}$ is to make the transformation be close to linear near the middle of the data, and have the "warping" part be farther out. The reason for the scaling factor f is so that 3dAllineate can restrict the search for C_{pqr} elements to the range $[-0.2, 0.2]$ (macro SETUP_BILINEAR_PARAMS).

Multiplying this out, we have

$$y_i + f C_{jik} (x_k - \Delta_k) y_i = A_{jk} x_k + b_j$$

or

$$y_i - f C_{jik} \Delta_k y_i - b_j = [A_{jk} - f C_{jik} \Delta_k] x_k$$

or

$$A_{kj}^{-1} y_j - f A_{kj}^{-1} C_{jik} \Delta_k y_i - A_{kj}^{-1} b_j = [\delta_{jk} - f A_{kj}^{-1} C_{jik} \Delta_k] x_k$$

so again we have $\tilde{C}_{jik} = -A_{jk}^{-1} C_{jik} \Delta_k$ $\tilde{b}_j = -A_{jk}^{-1} b_k$

but $\tilde{A}_{jk} = A_{jk}^{-1} - f A_{jk}^{-1} C_{jik} \Delta_k$

$$\tilde{\Delta} = \underline{0} \quad \tilde{f} = f$$

[Note that $\underline{\Delta}$ is ambiguous, as discussed on the next page]

General Bilinear Warp and Reduction to Standard Form

(4)

A general warp is

$$y_i = \left[\left[E_{pq} + F_{pqr} x_r \right] \right]_{ij}^{-1} \left[G_{jk} x_k + h_j \right] + t_i$$

or

$$\begin{aligned} E_{ji} y_i + F_{jik} x_k y_i &= G_{jk} x_k + h_j + E_{ji} t_i + F_{jik} t_i x_k \\ &= (G_{jk} + F_{jik} t_i) x_k + [h_j + E_{ji} t_i] \end{aligned}$$

Multiply both sides by E_{lj}^{-1} then interchange $l \leftrightarrow j$ to get

$$y_j + E_{je}^{-1} F_{eik} x_k y_i = E_{je}^{-1} [G_{ek} + F_{eik} t_i] x_k + [E_{je}^{-1} h_k + t_j]$$

Compare to \textcircled{A} to read off the "standard" form elements

$$\begin{aligned} b_j &= E_{je}^{-1} h_e + t_j & A_{jk} &= E_{je}^{-1} [G_{ek} + F_{eik} t_i] \\ C_{jik} &= E_{je}^{-1} F_{eik} \end{aligned}$$

Ambiguity in Δ and standard form

$$y_i = \left[\left[\delta_{pq} + f C_{pqr} (x_r - \Delta_r) \right] \right]_{ij}^{-1} [A_{jk} x_k + b_j]$$

is the general case with

$$E_{pq} = \delta_{pq} - f C_{pqr} \Delta_r \quad F_{pqr} = f C_{pqr} \quad G_{jk} = A_{jk} \quad h_j = b_j \quad t_j = 0$$

So reduction to standard form is clear

Two functions for manipulating bilinear warps are needed

① inversion (p.2)

② reduction to standard form

- taking 3dAllineate's output and standardizing it for inversion
- cat'ing with affine warps (next page)

Catenating Bilinear with Affine

(5)

Call the bilinear warp $\underline{W}(\underline{x})$ and the affine warp $\underline{L}(\underline{x}) = \underline{D}\underline{x} + \underline{e}$

$$\begin{aligned} \textcircled{1} \quad \underline{W}(\underline{L}(\underline{x}))_i &= \left[\left[S_{pq} + f C_{pqr} (D_{rs} x_s + e_r) \right] \right]_{ij}^{-1} \left[A_{jk} (D_{kl} x_l + e_k) + \right. \\ &= \left[\left[(S_{pq} + f C_{pqr} e_r) + (f C_{pqr} D_{rs}) x_s \right] \right]_{ij}^{-1} \\ &\quad \left. \left[(A_{jk} D_{kl}) x_l + (A_{jk} e_k + b_j) \right] \right] \end{aligned}$$

reduction to standard form obviously applies directly, after forming the relevant tensors

$$\begin{aligned} \textcircled{2} \quad \underline{L}(\underline{W}(\underline{x}))_i &= D_{iel} \left[\left[S_{pq} + f C_{pqr} x_r \right] \right]_{lj}^{-1} \left[A_{jk} x_k + b_j \right] + e_i \\ &= \left[\left[(S_{pq} + f C_{pqr} x_r) D_{qs}^{-1} \right] \right]_{ij}^{-1} \left[A_{jk} x_k + b_j \right] + e_i \\ &= \left[\left[D_{ps}^{-1} + f D_{qs}^{-1} C_{pqr} x_r \right] \right]_{ij}^{-1} \left[A_{jk} x_k + b_j \right] + e_i \end{aligned}$$

and again, reduction to standard form applies with \underline{D}^{-1} in the role of \underline{E} , etc.